TU Delft

Delft University of Technology

Bi-temporal foundation for LADM v2

Fusing event and state based modelling of Land administration data 2D and 3D

Thompson, Rodney; van Oosterom, Peter

DOI
10.1016/j.landusepol.2020.105246

Publication date
2021

Document Version
Final published version

Published in
Land Use Policy

Citation (APA)
Thompson, R., & van Oosterom, P. (2021). Bi-temporal foundation for LADM v2: Fusing event and state based modelling of Land administration data 2D and 3D. *Land Use Policy*, *102*, Article 105246. https://doi.org/10.1016/j.landusepol.2020.105246

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

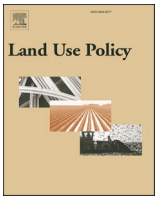
This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Land Use Policy 102 (2021) 105246

Contents lists available at ScienceDirect

Land Use Policy

journal homepage: www.elsevier.com/locate/landusepol



Bi-temporal foundation for LADM v2: Fusing event and state based modelling of Land administration data 2D and 3D

Rodney Thompson [a,*], Peter van Oosterom [b]

[a] 39 Salstone Street Kangaroo Point, Brisbane, Australia
[b] Delft University of Technology, Julianalaan 134 2628 BL, Delft, the Netherlands

ARTICLE INFO

Keywords:
3D cadastre
3D geoinformation
Land administration
Temporal
Bi_temporal

ABSTRACT

The prime purpose of Cadastral data – whether in the form of maps, survey plans or notes, or a digital database is the definitive demarcation of the extent of properties – and can be seen primarily as a decision support facility ("Can a structure be built here?", "Where can I build a fence?", "Should I buy this property"?). There are, however many additional uses for which this information has been applied – such as a base for the recording of assets such as light poles, underground cables, etc. and as a history of the pattern of land use and subdivision. Although secondary, these uses are important, and should be adequately supported including the historic information.

It is a fact that the determination of cadastral boundaries can only be carried out to a certain accuracy, and that that accuracy has been improving over time. Older surveys had been carried out with limited positional control, and using equipment with a low intrinsic accuracy by modern standards, although they correctly represent the topology between properties. As a result, later surveys provide an opportunity to improve the positioning of existing boundaries data without disturbing the topology of the existing data. In addition, engineering works such as road building, can provide a source of high accuracy position data that can be applied to improve low accuracy existing data.

This argues that the accuracy of boundaries should be improved in the historic record of the cadastre – after all we would like to see our historic parcels in the position we now know them to have been, so that they are comparable with current boundaries. Likewise, we need to correct inaccuracies in the attributes of the spatial objects and the topology between them (e.g. which spatial units are adjacent to or near a given object).

On the other hand, we must not lose sight of the decision-making side of the requirements – so that a past decision can be reviewed in relation to the data as it existed then. If the current knowledge in the database of today is used to review old decisions, they may seem irrational.

Data custodians are well aware of this issue, using terms like "update" to indicate a "real-world" change, while using "upgrade" to indicate an improvement of the database representation not accompanied with a change "on the ground"; however database software has not carried this knowledge through – resulting in its loss.

This argues for a database with bi-temporal history – where our current best knowledge of the history of the cadastre is recorded, and that history is corrected and maintained, while our past knowledge of the data also recorded as an audit trail (so that we can ask questions like "what did we in 2017 think the definition of this property was in 1994?"). This is realized via two types of time: database (or system) time and real world (or valid) time.

The different historic records, combined with changes of datum, can lead to confusion in terminology – where words such as "point", "position", "boundary" become overloaded. This paper is intended to provoke discussion of terminology to clear up this confusion, and potentially to assist with an extension of the temporal model as input for the revision of LADM to accommodate bi-temporality.

* Corresponding author.
E-mail addresses: rodnmaria@gmail.com (R. Thompson), P.J.M.vanOosterom@tudelft.nl (P. van Oosterom).

https://doi.org/10.1016/j.landusepol.2020.105246
Received 30 June 2020; Received in revised form 20 October 2020; Accepted 18 December 2020
0264-8377/© 2020 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Background

Cadastral data has some characteristics that are different from more conventional Geographic Information. The mixture of 2D and 3D real estate objects is the most obvious, but also the nature of the boundaries of objects is significant. “It has been pointed out repeatedly that only few objects in geographic space have natural boundaries which are sharp and well determined (Couclelis, 1992). Most geographic objects seem to be an abstraction of things which have unclear, fuzzy boundaries, if they have boundaries at all. The list includes most natural phenomena, from biotope to mountain range; extensive research efforts centre around soil type data (Burrough, 1986) and often use the techniques of fuzzy logic (Zadeh, 1974). Nevertheless, many practically used GIS model reality in terms of crisply delimited objects. Cadastral systems, GIS used for facilities’ management and automated mapping (AM/FM) and communal information systems all are appropriately oriented towards distinct objects with well defined boundaries. The same systems, with the same models are also used to manage soil maps and land use data, where the fiction of sharp boundaries contrasts with our view of reality” (Frank, 1995).

In real-world terms, cadastral boundaries are primarily sharp” fiat” objects, but some are less clear but” tangible” objects such as river banks (Smith, 1995; Thompson, van Oosterom et al. 2019). In all cases, the database representation is in terms of points, lines, surfaces etc. that are sharply defined. At best these definitions are qualified by accuracy metadata, however the action of changing the coordinate values of a” point” in the definition of a cadastral peoperty can be the result of any of nine different identified cases (identified in Section 2.5), with different results for current and historic information. Where changes of attributes and/or topology are involved, they may be caused by true events, or database correction, however on occasions it is difficult to separate the reasons for individual changes.

In Cadastral databases, the temporal aspect is critical – it is clear that the timing of the change of the shape of a spatial unit relative to the change of ownership of that spatial unit could be the subject of litigation. Often history is maintained only by archiving versions of the whole database once a year (or at regular or irregular intervals). This is unsatisfactory because whenever there is a change in database storage model or technology, either the history becomes inaccessible, or all of the multiple copies of the database must be converted. For this reason it is highly advisable to maintain the history within the live database (Sweetkind-Singer, Larsgaard et al. 2006; Janée, Mathena et al. 2008; McGarva, Morris et al. 2009).

For pragmatic reasons, where a cadastral database has a temporal aspect recorded, it is often a simple uni-temporal approach, where the changes are recorded using the” versioned object” pattern (van Oosterom, 1997, Tarbit and Thompson, 2006).

The disadvantage of this approach is that mistakes in the database remain for all time. For example, when an encoding error is discovered and corrected, an enquiry as at a point in time prior to the correction will return the error. An extreme example of this occured in the Queensland DCDB (Digital Cadastral DataBase). In the early days of capture, the structure permitted overlaps and gaps between parcels. The software allowed these innacuracies, but would not permit new cases to be introduced. By this strategy, a full cadastral database was made available several years earlier than if a” correct or nothing” approach had been enforced. Within a few years, the DCDB was made topologically correct and remains so, but any program that queries the data as at a date within those early years must be capable of processing topologically impure data.

There has been an effort of recent years to include bi-temporal data within the relational database model (Jensen, Clifford et al. 1994; Snodgrass, Böhlen et al. 1998; Kulkarni and Michels, 2012). Thompson et al., 2019a, 2019b suggest a bi-temporal object (versioned), which is changed via an event-based model. This results in a model that integrates state-based and event-based modelling.

Section 2 introduces the topic, presents some issues in the modification of cadastral object representations, and re-states some existing definitions. Section 3, is restricted to point-like objects in 2D, and suggests some basic terms, presents some issues in database representation, and re-visits the database transactions (update and upgrade) introduced in Section 2. Section 4 considers the concept of bi-temporal history as a technique to solve these issues, and as a concept relating to the LADM with regard to the upcoming second edition. Some issues raised by the extension of these into three spatial dimensions are addressed in Section 5, with conclusions and suggested future research in Section 6.

2. Application to Cadastre

2.1. The concept of the point

It may be a matter of surprise that the definition of a point is quite a complex issue. ISO19107 defines a point as a “0-dimensional geometric primitive, representing a position” (ISO-TC211, 2003 Page 10) and introduces the concept of “Direct Position” which defines the coordinates of a point in space. It does not restrict this to 2 or 3 coordinates, so could presumably be in terms of (x, y, z, t)[1] . A Direct Position is related to a Coordinate Reference System (CRS) (directly, via an enclosing geometric object or using default defined for the whole database), This is then used to define the GM_Point class, which is the simplest geometric primitive (GM_Primitive), and the basis of all other primitives. This is an apparently simple approach, but can lead to confusion in the Cadastral database field. Some further discussion is needed.

The Coordinate Reference System (CRS) specifies how the coordinates are to be interpreted. For the purpose of this discussion CRS is effectively equivalent to “Spatial Reference Framework” (SRF) and “Spatial Reference System” (SRS), with the latter term being used here. A series of standard SRS have been defined and given Spatial Reference Identifiers (SRID) to facilitate exchange of spatial data.

Several kinds of “point” exist in a Cadastre: the parcel corner, labelling points, survey marks, transit points, centres of circular arcs, etc.

2.2. Linear features

Most linear features in a 2D Cadastre are defined as a straight line between two points. Examples – a surveyed right line boundary of a suburban spatial unit, a recorded measurement of the bearing and distance between two points, even the bank of a river that defines a natural boundary is frequently represented as a connected series of straight lines. Counter examples include various curves which are defined by references to points (e.g. an East-West boundary will actually be a curve known as a loxodrome). Some cadastres also allow mathematically defined curves such as a segment of a circle.

2.3. Area features

In a 2D Cadastre, most spatial units are area features defined by their boundaries, which in themselves are linear features as described above (Section 2.2).

In addition, it is common for a spatial unit to have one or more labelling points defined for the purpose of identification. While the location of these points does not usually need the same accuracy of recording as actual cadastrally significant points, they are required to be inside the area feature. It is important that all constraints on the definition of area features are preserved (e.g. boundary lines may be constrained to touch only at end points) as accuracy is improved.

[1] To save excess verbiage, unless otherwise noted, in this paper (x, y, …) is taken to include geographic coordinates – with longitude as x, and latitude as y.

2.4. Boundary surfaces

In a 3D Cadastre, where geometric definitions of spatial units are recorded, they are typically by reference to a boundary (B-rep or Boundary Representation). A 3D spatial unit, in this representation is defined as being bounded by a number of boundary surfaces. There are some important constraints that must be observed. For example, if a surface patch of a volumetric spatial unit is required to be planar, then the modification of the defining points of the surface must not violate this requirement – although a tolerance may be allowed (Stoter and van Oosterom, 2006). There is also typically an important constraint that a 3D spatial unit is enclosed in a 2D base spatial unit.

2.5. Database modifications

Consider a (3D) spatial object *O* which is modified in the database to a new location and possibly a new shape *O'*. This action may be categorised as:

1 **"Tangible Movement":** The object may have been physically moved a small distance on the ground (*O* to *O'*) – for example a survey control point may be accidently moved by roadworks.
2 **"Correction":** The recorded position and/or shape of *O* may be found to be wrong, and it is corrected to *O'*.
3 **"Natural Movement":** An ambulatory boundary may be defined by the location of a natural feature, and the boundary representation may be moved each time the feature is observed.
4 **"Datum Change":** There may be a change of SRS throughout the database. The values of all coordinates have changed, but using the new SRID, the same physical locations should be determined (any point *p* in the old SRID should equate to *p'* in the new as a real-world location). There should be little or no change in the shape of any object representation.
5 **"Dynamic Datum":** The coordinates may be referred to a Dynamic Datum (Donnelly, Crook et al. 2015) (see Section 3.1), in which case the coordinates of all points which are fixed to the ground (such as a cadastral corner) are changing on a daily basis with continental drift.
6 **"Local Deformation":** There may be a local or semi-local constant and predictable movement of the earth's surface not associated directly with continental drift. Some examples are subsidence or upthrust and soil creep. This may distort actual shapes of objects.
7 **"Unanticipated Deformation":** There may be a local event such as an earthquake, landslide, etc. that requires the re-positioning of many points, and redefinition of shapes over a local area. This event is assumed to be unanticipated and unpredictable.
8 **"Topology Preservation"** It may be necessary to insert a new vertex in a spatial unit boundary because the neighbour unit has been subdivided. In 3D, a face may need to be split by an edge for the same reason.
9 **"Surface Curvature"** Where lines and faces are fairly large, they may have been approximated as a straight line, whereas in fact they follow the curve of the Earth's surface. If the accuracy of the data is being improved, this approximation may not be acceptable, and an improved representation may be needed (e.g. introducing points, edges, or using a parameterised curve). This should affect history because the line/face has always been curved.

The effect on the history of the Cadastre varies depending on the reason for modification of a direct position coordinates, so to clarify the situation, some terminology is proposed.

2.6. Externally defined terms

Terminology is, in general, taken from "ISO19103: Geographic Information – Spatial Schema" (ISO-TC211, 2003), but the following additional terminology has been adopted:

Spatial Unit: A "Single area (or multiple areas) of land … and/or water, or a single volume (or multiple volumes) of space" (ISO-TC211, 2012 Page 6).

Fiat Object: "exist only in virtue of the different sorts of demarcations effected cognitively by human beings" (Smith, 1995 Page 477).

Tangible Object: (Smith uses" Bona Fide")" exist independently of human cognitive acts" (Smith, 1995 Page 477).

Transaction Time: "time when a fact is current in a database and may be retrieved" (ISO-TC211, 2002 Page 6).

Valid Time: "time when a fact is true in the abstracted reality" (ISO-TC211, 2002 Page 6).

Instant:" point representing [a] position in time" (ISO-TC211, 2002 Page 4)

Direct Position: "… hold the coordinates for a position within some coordinate reference system." (ISO-TC211, 2003)

3. Proposed terminology

3.1. Point-Like objects in 2D

The first part of this discussion considers only points with two spatial dimensions – assuming a surface with no topography (i.e. a plane or the surface of a spheroid), and no 3D spatial units to consider. It is assumed that there exists an International Terrestrial Reference Frame (ITRF), which in effect embeds a set of axes in the Earth, and defines a set of polar coordinates (Latitude, Longitude) that do not change as the continents drift or other deformations occur. In such a reference system, a cadastral spatial unit's coordinates will be constantly and steadily changing. It should be noted that "ITRF, which is sometimes described as a 'dynamic datum', is in reality a static reference frame with kinematic coordinates for ground-fixed physical features" (Donnelly, Crook et al. 2015 Page 237). Bearing this in mind may save some confusion in the following discussion, but "dynamic datum" is in common parlance, and will be used here.

Constantly changing coordinate values would be inconvenient for day-to-day usage in a Cadastre, so a Local Reference Frame (LRF) is used. This remains fixed relative to local tectonic plate movements (at least to the accuracy we need). It is expected that a set of coordinate transformations exist such that if $p = p_t(x, y, t)$ is the ITRF representation of the 2D location of the point at instant t, and $p{=}p_l(x', y', t)$ is the same location represented in an LRF at the same instant of time, that functions such as:

$$x = x_{lt}(x', y', t),\ y = y_{lt}(x', y', t)$$

$$x' = x_{tl}(x, y, t),\ y' = y_{tl}(x, y, t)$$

which can provide a two-way transformation between the two frameworks.

This leads to a suggested set of point-like objects for discussion:

3.1.1. EarthFixedLocation

This is a location which remains the same position relative to the ITRF over time. (No assumption of any object occupying the position, or any measurement of the location having been made).

3.1.2. PlateFixedLocation

This retains the same position relative to the local tectonic plate over time. (No assumption of any object occupying the position, or any measurement of the location having been made). An example is the location of a Cadastral corner. Viewed in relation to an ITRF, this ceases to be a point and becomes a trajectory.

3.1.3. PointInstant

a tuple of coordinates (referred to an SRS) of a point at an instant of time. Coordinates would be (x, y, z, t).

3.1.4. DirectEarthPosition

By analogy with DirectPosition defined in ISO19107, this is a tuple of coordinates (e.g. lat, lon) referred to an ITRS as mentioned above.

3.1.5. DirectPlatePosition

This is a tuple of coordinates referred to a LRF (based on this tectonic plate).

3.1.6. DirectPlateTrajectory

Where a PlateFixedLocation is referred to an ITRS, this is not a point, but a trajectory of PointInstants (x, y, z, t) indicating the movement of the PlateFixedLocation. See Fig. 1.

Thus a cadastral corner is an instance of a PlateFixedLocation, located by a DirectPlatePosition (static datum), or a DirectPlateTrajectory (dynamic datum). But this is only true for a period of time. Once modifications to the real-world objects and database representations are permitted, more objects need to be considered.

3.1.7. PlateFixedPoint

A real-world location that is fixed relative to the continental plate. It may be the location of a tangible object – such as a survey mark, or a fiat object such as a node along the boundary of a spatial unit.

3.1.8. EarthFixedPoint

A real-world location that is fixed relative to the Terrestrial Reference Frame. It is unlikely to be the location of a tangible object (unless for an instant only), It might be a fiat object, such as the intersection of the Equator with the International Date Line.

3.1.9. Point

In this paper this term is used as a locator for cadastral corner, survey point, etc – any PlateFixedPoint that is of interest to the cadastre. A point cannot move relative to the plate – if a cadastral corner is moved relative to the Earth's surface, it must be associated with a new Point, creating a new version of the cadastral corner.

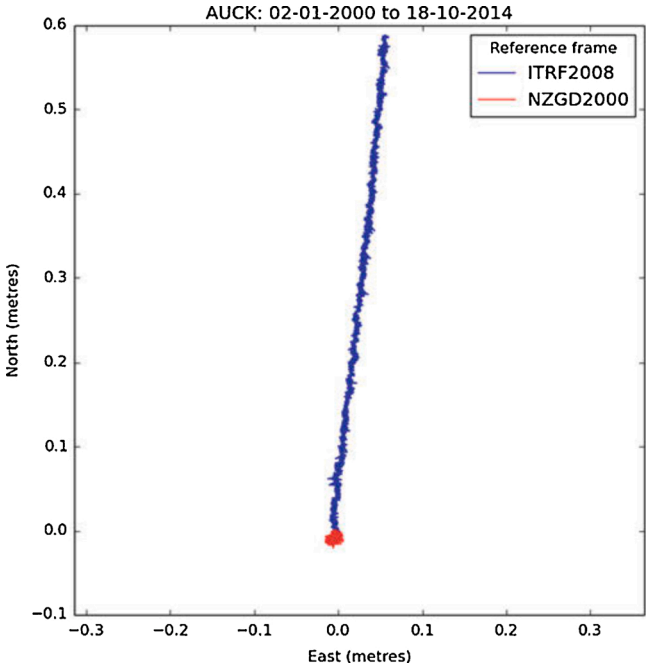


Fig. 1. Example of the trajectory of a PlateFixedLocation (a reference point in Auckland New Zealand) over a 15 year period. The position in relation to an ITRF is blue and red in relation to an LRF (NZGD2000). Note that "The trajectory in terms of the local frame is almost static". (Reproduced from Donnelly, Crook et al. 2015 Page 239). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. Database representation of point-like objects

As described in Section 2.1, a Direct Position (and therefore the coordinates of all points and other geometric primitives) must be related to an SRS (by an SRID). One possible approach is that each representation carries the SRID – so that $p = p(x, y, z, s)$ or $p(x, y, z, t, s)$, where s is the SRID used to interpret the coordinates, and t is the date/time that the coordinates were determined, and is needed if a dynamic datum is involved.

ISO19107 also permits the SRID to be specified via an enclosing geometry, so it could be, for example be recorded on each Spatial Unit, and apply to each point in the definition of the boundary. It could also be specified at the city, state, or country level in an extended database.

Commonly, a single SRID is defined for the whole database, with all incoming data being converted to that frame. If a non-dynamic SRS is in use, it will be necessary from time to time (typically a few decades apart) to convert to a new SRS. This can be a sweep through the database applying a set of functions. While this is not difficult *per se*, it may lead to topological errors in some database structures.

However these are not the only possibilities. For example it may be decided not to back-date a change of SRS to historic geometries applying it to current data only, so that point positions time-stamped before a certain date are in an older SRS, while after this date a new SRS is in use.

3.3. Movements revisited

Returning to the cases identified in Section 2.5:

1 **Tangible Movement:** The actual object is moved relative to the earth surface. New Points must be created, and the object is linked to the new Points. This means that the history of the object will show a real-world jump in the position at this instant of time. This will affect any observations of (say) cadastral corners that are fixed in reference to the mark in the future, but should not affect any historic points referred to it. In Fig. 2B, the new plate fixed point p' is created, and the cadastral corner moved to it.
2 **Correction:** The recorded position is corrected. This is effected by a change to the coordinates of the points defining the object. The history of the object or objects will subsequently show that they have always been in the adjusted position, (but see Section 4 on the bi-temporal issues involved). In Fig. 2C, there is no change to the real-world situation, but the coordinates on p will change.
3 **Natural Movement:** The point associated with a natural ambulatory boundary. This is treated as in case 1, but with the knowledge that the point location as surveyed does not have enduring legal status. This is a case of a PointInstant, because it is valid only for the instant of its measurement. Any future locations will depend on its being revisited.
4 **Datum Change:** This has been partially discussed in Section 3.2, but in terms of the real world objects, the Points are adjusted to the new coordinates (and link to the SRS). The objects linked to the points are not affected because their position has not been changed. The Point records will carry history of the change, but the cadastral corners, marks, etc will not.
5 **Dynamic Datum:** Where coordinates are referred to a Terrestrial Reference Framework. This requires a set of coordinate velocities to be determined for each point. The Point records will probably not change, and the corner, mark etc records will certainly not. In Fig. 2D, this is shown relative to an TRF – the points, corner, boundaries, and LRF lines have all moved. This is discussed further in Section 3.4.
6 **Local Deformation:** Local or semi-local constant and predictable movement of the earth's surface. This is probably seen as a distortion of the LRF, or a variation of the velocity field – see Section 3.4. If it is

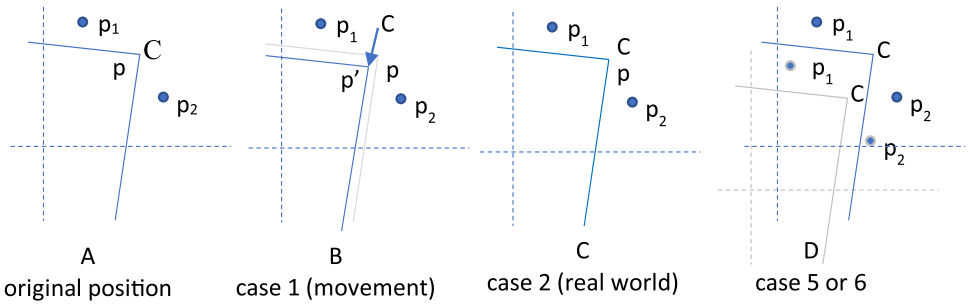


Fig. 2. Movements of a cadastral corner under various changes. Points p_1 and p_2 are assumed to be physical markers near the corner C. The location of C is marked as p. The two dashed lines indicate a local reference frame coordinate values.

not recognised as an effect, it will be (spuriously) identified as case 2, or even worse as case 1.

7 **Unanticipated Deformation:** Such as earthquake. All cases differ in extent and effect, but many new points will have to be created, and cadastral corners re-linked. Careful judgment is needed, and survey marks and such landmarks will have to be reviewed and replaced as needed. It would be unlikely that Point records would be moved – and so there would be no back dating of their positional information. The actual cadastral database objects would be affected, and carry bi-temporal historic record of the event.

8 **Topology Preservation:** Insertion of a new vertex or line in a spatial unit boundary because the neighbour unit has been subdivided. This can be effected by the creation of a new point / line – but in fact, the point / line has always existed along the boundary, it simply hasn't been recognised as important in the past (Fig. 3).

9 **Surface Curvature:** If the accuracy of the data is being improved requiring the creation of new points along the linear feature, or parameterising of the curve, it should be observed that the real-world linear feature has not changed, and the newly defied feature was always part of the cadastre (Fig. 4).

3.4. Dynamic datum

As mentioned in Section 3.1, this is a confusing term, but being in general use it is adopted here. In effect, all the point-like objects in the cadastre (such as boundary corners, traverse points, survey marks …) are recognised to be PlateFixedPoints, while the coordinate system is based on an ITRF. Thus a Point will be represented by a DirectPlate-Trajectory, meaning that the question "What is the Latitude and Longitude of the corner of my property?" cannot be answered except as a rough approximation. The question needs to be qualified by "at date …".

To provide an answer to this question, there are a number of possible solutions. One is that each Point records the velocity of the plate at that location – so that $p = p(x_t, y_t, z_t, v_x, v_y, v_z)$ records the point location at time t. In order to find the coordinates at date t', $x = x_t + v_x(t'-t)$, $y = y_t + v_y(t'-t)$, $z = z_t + v_z(t'-t)$. The values x_t, y_t, z_t and t are obtained from the SRS, which is located using any of the methods discussed in Section 3.2. This method has little to recommend it – the maintenance problems are obvious, as is the effect of a mis-recorded velocity.

At the opposite extreme – a small jurisdiction within a stable continental plate may elect to store just one value for (v_x, v_y, v_z) for the whole cadastre. Note that the Australian plate is very stable, but one of the faster-moving, while New Zealand, being at the edge of a plate, is slower moving but more complex.

Fig. 3. Topology Preservation: Where a spatial unit B is subdivided, a point representation p is created, although in fact, the real-world point has always existed along the boundary between A and B. A finite representaion of p, however will almost certainly not lie on the line, and the introduction of p will cause a slight bend.

Various approaches to modelling the motion of the tectonic plate have been used, including block moves, distortion grids, n-parameter transformation, the vector TIN (Triangulated Irregular Network) (ICSM, 2009; Donnelly, Crook et al. 2015; ICSM, 2018; Thompson and Van Oosterom, 2019; ICSM, 2020)

4. Bi-temopral history

In the bi-temporal history model, two forms (or dimensions) of time are involved. One is the real-world time (known as "valid time") and the other is the time a piece of information enters the database (known as "transaction time") (Snodgrass, Böhlen et al. 1998). This can be accommodated by two types of event and three abstract classes (Fig. 9):

ValidEvent: An event which happens in the real world. The date/time in this object is the actual time the event happened. (Although well accepted, the term "valid" is unfortunate, because not only does it imply some kind of validation operation, it creates the impression that TransactionEvent is less "valid").

TransactionEvent: The entry or modification of some data in the database. In this case, the date/time is supplied by the computer system.

Lifespan: "The lifespan of a database object is the time over which it is defined. The **valid-time lifespan** of a database object refers to the time when the corresponding object exists in the modelled reality, whereas the **transaction-time lifespan** refers to the time when the database object is current, in the database." (Jensen, Clifford et al. 1994 Page 55).

To include the temporal aspects in a Relational Database, the following three definitions are defined by Jensen, Clifford et al. (1994 Page 54). Note that in this context, "relation" refers to the database table (or object class) and not to the relationship between two tables (or classes), while the term 'tuples' is used for records in the table (or instances of the class):

ValidTimeRelation: "A [database] relation with exactly one system supported valid time. There are no restrictions on how valid times may be incorporated into the tuples; e.g., the valid-times may be incorporated by including one or more additional valid-time attributes in the relation schema, or by including the valid-times as a component of the values of the application-specific attributes." (That is to say the time stamps may be administered by the database schema or by the application, and may be of granularity appropriate to the application: i.e. at tuple/record granularity or at attribute granularity). For consistency with the LADM – as an analogy with LA_VersionedObject, the term ValidVersionedObject will be used here.

TransactionTimeRelation: "A relation with exactly one system supported transaction time. As for valid-time relations, there are no restrictions as to how transaction times may be incorporated into the tuples." For consistency with the LADM the term

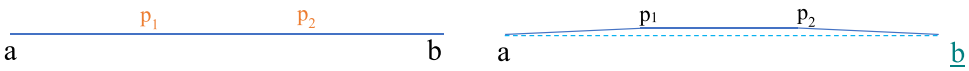


Fig. 4. Surface Curvature: The long boundary a-b follows a line of latitude. In the past, a straight line was considered acceptable, but with improvements in measurements intermediate points p1 and p2 have been inserted. (They have always existed along the boundary, but line a-b has been inaccurately been represented as a straight line).

TransactionVersionedObject will be used here.

BitemporalRelation: "A relation with exactly one system supported valid time and exactly one system supported transaction time. As for valid-time relations and transaction-time relations, there are no restrictions as to how either of these temporal dimensions may be incorporated into the tuples." For consistency the term BitemporalObject will be used here.

Analogously to the spatial reference frameworks, there exists an international standard temporal framework UTC (Coordinated Universal Time or Zulu Time), and many local frameworks. This is beyond the scope of this paper, which assumes that a suitable framework has been chosen for the database, and a computer system date/time with a fine granularity is available.

Both forms of time are discretely modelled in the database as taking effect at specific instants, apart from the datum movements, which are continuous. Where dynamic datum effects are to be considered, the changes in coordinate values take place in valid time, and therefore can be applied to historic or future data.

4.1. Database support, state based history

Bi-temporal support has been included in the standard SQL11 (using the terminology Application_Time, and System_Time) (Kulkarni and Michels, 2012). Temporal support is included in several database systems: IBM DB2 (Saracco, Nicola et al. 2013); Oracle (Jernigan, Guo et al. 2009); and is available in PostgreSQL (Clark, 2015) although the implementations differ in nomenclature and details from the proposed SQL support. ESRI's Parcel Fabric has attributes "System Start Date" "System End Date" "Legal Start Date" and "Legal End Date" on the Parcels and Points tables (ESRI, 2016). As described, this approach can be termed "state based" – since it permits the determination of the state of the data as at a date/time in the past.

The Land Administration Domain Model (LADM – ISO19152) (ISO-TC211, 2012) partly provides bi-temporal support. It specifies an abstract VersionedObject class with time stamps – covering the TransactionTime concept (beginLifespanVersion and endLifespanVersion, which correspond to the relevant LA_Source lifeSpanStamp attributes); while LA_Source can also record the valid date/time of the linked objects, but currently there is no corresponding valid time interval/lifespan in VersionedObject. For example, all changes that affect the spatial units in the cadastre should be recorded using a LA_SpatialSource object (an instance of this subclass of LA_Source), which is directly linked to the geometric and other versioned objects by means of the lifeSpanStamp attribute "The moment that the event, represented by the instance of LA_Source, is further processed in the LA system (this is the moment of endLifespanVersion of old instances, and the moment of beginLifespanVersion of new instances" (ISO-TC211, 2012 Page 15). In addition, the LA_RRR table can have an additional time specification indicating when this is supposed to be valid (this can be reoccurring patterns, or an end date in the future; e.g. a long lease ending after 50 years), which influence the actual relationship between spatial unit and party.

4.2. Archiving historical data

It has been the practice to archive historical information to a backing storage device, to save space and improve responsiveness of database which supports enquiries on current data only. This is not being suggested here – the problems of keeping archived data in a useful and accessible form mitigate against this (Sweetkind-Singer, Larsgaard et al. 2006), and the VersionedObject approach has proven to maintain acceptable performance even as the data volumes increase due to the build-up of history. (In the Netherlands Kadaster, and also in the Queensland Cadastre, the growth rate is such that the raw size doubles in about 7–8 years, but the spatial indexing search time is O(log n) in both cases, so the slowing in response is hardly noticeable). The suggestion being made here is that the historical versions of the database tuples are retained in the database for all time.

4.3. Benefits of the bitemporal model

In order to illustrate the benefit of a bitemporal model in the storage of cadastral data, consider the alternative. The Queensland (Australia) cadastral database has been operating with a uni-temporal model since 2001 (Thompson, 2015), and thus has nearly 20 years of history available Fig. 5. The uni-temporal model was chosen as giving a relatively easy data capture process, and being cheap to implement. While the approach has merit, the disadvantages (relative to the bitemporal model) can be seen in the following figures.

As can be seen in Fig. 6, there are additional parcels (a), a change in the definition of the river boundary (b and c), and an adjustment of parcel boundaries (d). It is not easy to determine if the recorded change in the river's edge (e and f) has been the result of improving the measurement (TransactionEvent), or whether it is due to the erosion and accretion of the natural boundary (ValidEvent); and whether the parcel boundary changes (d) were a result of real property boundary changes (ValidEvent), or re-measurement of the original boundaries (TransactionEvent). It is to answer these issues that the bi-temporal model is suggested.

Practical value is given to this where the cadastral boundaries are used as control for other jurisdictional data (e.g. street furniture such as underground conduits) (Priebbenow, 1993). One of the benefits of a bi-temporal model is that a real-world movement of a boundary (which will not affect the coordinates of linked street furniture), can be distinguished from the re-measurement of a boundary (which should lead to a re-calculation of linked object locations).

4.4. ACID (Atomicity, Consistency, Isolation and Durability) in history

The so-called ACID database concept is central to relational database design and construction, and it is important it is not lost when historic data is retrieved. Accordingly, when processing a transaction and storing new records, changing records or deleting records, all must get the same transaction time stamp (even if in reality a transaction has its own duration). When data is retrieved as at some time in history, the mechanism is that transaction time expressions such as Tmin $\leq$ time_hist $<$ Tmax are included in the selection clauses of all temporal tables (with the subtle assumption that Tmax is set to infinite when no end data/time is specified). That is to say all timestamps are quantised – such that all objects which are updated in a single transaction are stamped with exactly the same DateTime (van Oosterom 1997).

Consider the case of Fig. 7. Transaction 1 is being written to the database when Transaction 2 starts to load. Unless "dirty read" is

Fig. 5. A: Queensland Cadastre in 2001. B: Queensland Cadastre in 2013 (from Thompson, 2015).
Broad changes can be seen in the cadastre in Fig. 5 – parcel a (extending into the river) has been replaced by a more closely defined jetty structure b; jetty c has been captured, and many volumetric (purple) and strata (red) spatial units have been captured.

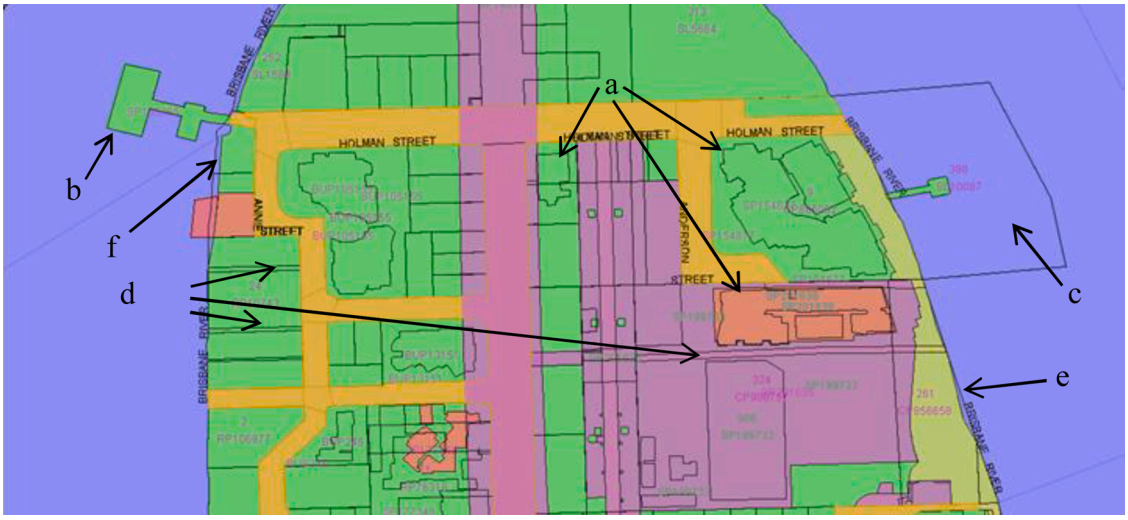


Fig. 6. Detail from 2001 superimposed over 2013. Note the change in definition of the river boundary (from Thompson, 2015).

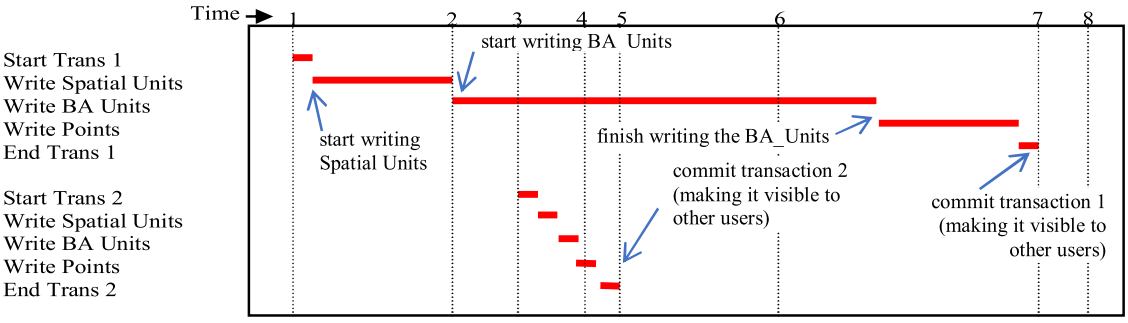


Fig. 7. Two transactions (affecting different spatial units) being committed to the database at overlapping times. Each horizontal bar represents the time taken to physical update the database, but only when the full transaction is completed (and committed) will other users see the updates.

enabled[2], someone querying the database will see:

- Querying at time 4 or earlier, they will see nothing updated.
- Querying at time 6, they will see transaction 2 complete, no effect from transaction 1
- Querying at time 7 or later they will see both transactions complete.

At no time will a user see a part completed transaction. This is important, because the references, including topology of the data may

[2] Some databases, for response reasons allow “dirty read” - which balances speed of access against the ill effects that partial transactions might cause. Software developers using “dirty read” must at all times be aware of the possibility of retrieving inconsistent data.

be invalid while a transaction is incomplete.

Now consider the case where each TransactionTime object is time-stamped at the transaction start time. An enquiry made after both transactions are complete, specifying an "as at" time will see:

- Querying as at time 2, will only retrieve results from transaction 1, including the points that define them, even though they were actually inserted later on.
- Querying as at time 4, they will see transaction 1 and 2 complete, despite that in reality both where not completely updated.
- Querying as at time 6, they will see transaction 1 and 2 complete, despite that in reality only transaction 2 was completely updated.
- Querying as at time 7 or later they will see both transactions complete (as was also the case of the query at time 7 in reality).

This is rather paradoxical, because, for example, an enquiry "as at" time 4 will see all of transactions 1 and 2 committed while an enquiry in real time at time 4 will not see either transaction committed[3] . The assumption is that that time intervals for the actual database transaction are relatively very short, normally sub-second. So, on the time scale of land administration, these transactions can be considered to have happen at a single moment.

4.5. Event based history

An alternate solution to state based-based modelling with much to recommend it is event-based approach. This uses the concept of an event, identified by an eventID. For now, consider the case of the transaction time history. Each time there is a transaction committed to the database, a surrogate eventID is created sequentially which becomes the primary key in the TransactionEvent table. This ensures that for any two events a and b:

a.eventID $>$ b.eventID $\Rightarrow$ a.systemDateTime $\geq$ b. systemDateTime

a.eventID $=$ b.eventID $\Rightarrow$ a. systemDateTime $=$ b. systemDateTime

a. systemDateTime $>$ b. systemDateTime $\Rightarrow$ a.eventID $>$ b.eventID

Note that in this strategy, it is possible for two events to be distinguished even though they have the same systemDateTime – i.e. they were committed within a shorter period than the granularity of the DateTime type. The major advantage of this strategy is that it provides a place (a TransactionEvent record) in which to store metadata of the update (who, why, etc.). There are two main options in classic event-based modelling with event information to reconstruct how to move from one state to the next state, without storing the states explicitly: forward or backward. Forward event-based modelling consists of the initial database records and the content of the event records allowing to go forward in time and arrive at the required moment/state. The drawback is that this is rather error prone, as one error or missing event breaks the chain. Also, most people want to access the current state and this will then imply a complete chain traversal and slow response. Backward chaining maintains the content of the current database, and the event records are reversed, going further and further back in time. However, backward chaining is also error prone and can be slow if one wants to go back rather deep in history. Backward chaining has advantages over forward in that if the chain is broken, history is compromised rather than the current data, and for enquiries on current data, no chain tracing is needed.

4.6. Event and state based history

In reality, state-based and event-based modelling can be combined well using the eventID in place of the timestamp in the TransactionVersionObject.

Referring to Fig. 9, this approach uses surrogate eventIDs for the attributes SysTmin, SysTmax. (This is the approach used in the Queensland Cadastral Data Base – which has transaction time history only). For example, the spatial unit record (a TransactionVersionObject) contains a "creating" and "retiring" event identifiers SysTmin and SysTmax (if there is no "retiring event" a SysTmax of "infinity" is stored). Thus the state of the database can be determined as at any point of time using the familiar "where clause" construct:

where SysTmin $<=$ eventID and eventID $<$ SysTmax

using the smallest eventID with timestamp $\geq$ the required date/time.

Thus the state-based requirement is accommodated, while the eventID makes accessible the metadata of the event that created and/or destroyed the version of the spatial unit.

Note that in LADM v1 both VersionedObject (as realization of state based modelling) and LA_Source (as realization of event based modelling) have the following attributes:

quality: DQ_Element [0..*]

source: CI_ResponsibleParty [0..*] (ISO-TC211, 2012)

which allow the recording of metadata both to the event and state. However, in LADM v1 there is no explicit relationship between VersionedObject and LA_Source. This should be improved in LADM v2 to better integrate event and state based modelling.

4.7. ValidTime events

While some of the (TransactionTime) issues discussed in 4.4–4.6 also apply to the ValidTime events, there are differences. Because there may be geographic parts of the cadastre which are more dynamic than others (for example, the update cycle is faster in an area of high development) it is likely that valid time events will arrive in the database in non-chronological order. In an extreme example, an important transaction may have been overlooked in the data entry process for several days. In the interim, many unrelated transactions may have been included in the database, but chronology of the real-world actions needs to be maintained. Thus a simple sequentially allocated eventID may not be appropriate. One possible solution is a combination of a dateTime stamp and a unique event identifier (see Future Research - Section 6.1).

Slotting in ValidTime events on a geometric object that already has later events can be difficult (See Fig. 8). It is essential that topology is maintained throughout the history, and that no update is accidently lost. For this reason, it may be considered desirable that no out-of-sequence ValidTime events are permitted to be applied to any individual ValidVersionObject – that is out of sequence events are possible in different parts of the database or different classes of object, but not on a single object.

As a counter-argument to this, several jurisdictions are in the process of back-capturing historic survey plans (Grosvenor, 2019). In this case, historic data will need to be entered out of sequence – while ensuring the "lost update" problem is avoided. Maintaining a valid topology while permitting out-of-sequence entry of updates is quite a difficult problem, but an alternative was proposed by Thompson (2015) whereby a different standard of database consistency is accepted for older historic items (as a function of the different levels of validation achievable in the past).

Another difference is that the granularity of the recorded time in ValidTime transactions is frequently coarser than TransactionTime. Although database timestamps may be recorded to a resolution of microseconds, it is unlikely that the recording of real-world events is to this accuracy. Most events like submission, registration etc. of plans of survey may be recorded as the date of the event only. This significantly raises the probability of multiple events having the same timestamp.

[3] Ideally the commit time of the transaction would be used for the time-stamping – so that transaction 1 would be timestamped at time 7, and transaction 2 at time 5, but this is difficult because it requires advance knowledge of how long the transactions will take to be written.

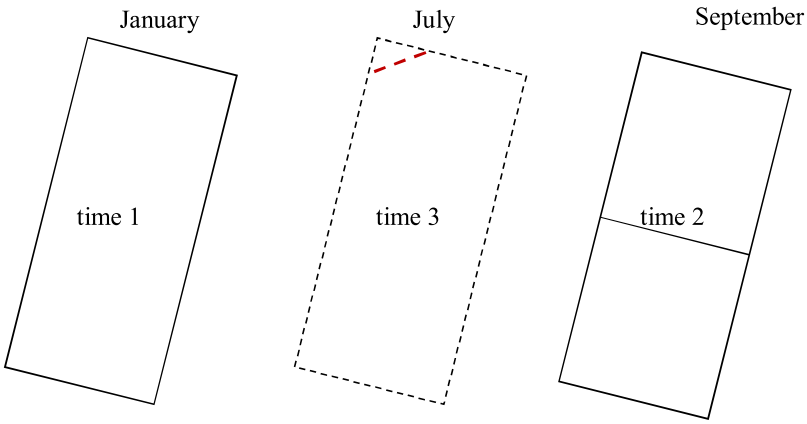


Fig. 8. This spatial unit existed as a rectangle in January, had a corner truncated in July, and was subdivided in September. The changes hit the database in a different order - the subdivision at transaction time 2, and the truncation at transaction time 3. As a result, the truncation is not shown as applying in September.

4.8. Bitemporal model

The state-based and event-based approach can be combined in an integrated model. It is suggested that the Event classes be real tables similar to the LA_Source class in LADM (carrying the event metadata for the database changes – as shown in Fig. 9). In this simplified model, SysTmin, SysTmax, Tmin and Tmax are all eventIDs and the actual date/time of these events are stored in the TransactionEvent / Valid event tables.

The time range classes in Fig. 9 (TransactionInterval, ValidTimeInterval and VersionedObject) are abstract. This means they provide min and max timestamps to their subclasses. For example, a class such as LA_Spatial_Unit that is a subclass of VersionedObject would carry the inherited attributes SysTmin, SysTmax, Tmin and Tmax, making it a" BitemporalRelation" using the terminology of Jensen, Clifford et al. (1994), or a BitemporalObject.

4.9. Example

This artificial example does not assume a specific database schema, but rather is a "pseudo-table" presentation – with only the temporal columns shown in detail. Consider the real world history of a spatial unit

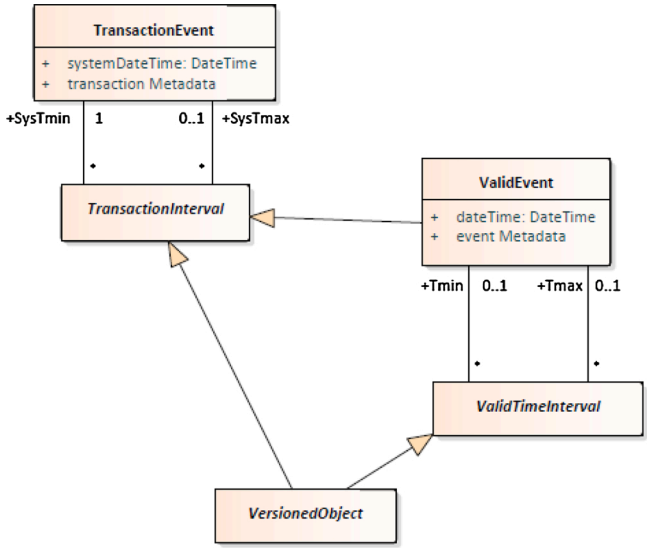


Fig. 9. Bi-Temporal concepts. ValidEvent and TransactionEvent are representations of a single action such as creating a parcel. The others are time range (abstract) classes, with VersionedObject carrying both intervals.

Table 1
Real World history.

unit ownership details	Tmin	Tmax
Owned by A	1960	1/Nov/2011
Owned by B (purchase for $3 m)	1/Nov/2011	null
valuation details	**Tmin**	**Tmax**
Valuation $900,000	1/Jul/2010	1/Jul/2011
Valuation $1m	1/Jul/2011	1/Jul/2012
Valuation $1.1m	1/Jul/2012	1/Jul/2013

(Table 1). These events seem surprising, given the mismatch between the purchase price ($3 m) and the valuation ($1 m) at the time. However, if a full bi-temporal history is available, a different, and darker picture emerges (Table 2).

Notice that during a period around the time of the purchase, the database had been modified to raise the valuation. Such as event would be the subject of investigation, and although this data might be recorded in a conventional modification log, the search would be difficult. This allows the information to be presented in an easily recognised form. Note – this is entirely a fictitious case, and would be unlikely to succeed in practice.

Where history is permitted to be corrected, the ValidEvent itself may be an object with DB history (implying that its dates and metadata can be modified/ corrected, resulting in multiple versions of the ValidEvent data). That is to say, the table of ValidEvents can be a TransactionInterval relation, i.e. inheriting from this superclass. (Other implementations are possible - See (Thompson, van Oosterom et al.

Table 2
Bi-temporal History.

unit ownership details	Tmin	Tmax	sysTmin	sysTmax
Owned by A	1960	1/Nov/2011	2000	3/Nov/2011
Owned by B (purchase for $3 m)	1/Nov/2011	null	3/Nov/2011	null
valuation details	**Tmin**	**Tmax**	**sysTmin**	**sysTmax**
Valuation $900,000	1/Jul/2010	1/Jul/2011	6/Jul/2010	4/Jul/2011
Valuation $1m	1/Jul/2011	1/Jul/2012	4/Jul/2011	30/Oct/2011
Valuation $3.5m	1/Jul/2011	1/Jul/2012	30/Oct/2011	3/Nov/2012
Valuation $1m	1/Jul/2011	1/Jul/2012	3/Nov/2011	5/Jul/2012
Valuation $1.1m	1/Jul/2012	1/Jul/2013	5/Jul/2012	null

2019)). Allowing update of the ValidEvent metadata itself – particularly if the date/time is modified – can cause difficult validation and constraint verification requirements. (See section 4.7).

The TransactionEvent itself should not have history, because the only option to change a transaction event should be to purge it and all its results, and resubmit them (and this can only be done if no later events have been recorded on the same base objects). It may well be, in any case, that the TransactionEvents are intended as an audit trail, in which case modifications would be inappropriate.

4.10. Implementation in the Land Administration Domain Model

As in the current LADM standard (ISO-TC211, 2012), VersionedObject is an abstract class, and provides begin and (optional) end Lifespan and Valid Timestamps (optional) to the inheriting classes (Fig. 10). The class VersionedObject is used in the LADM to manage and maintain historical data in the database. History requires that inserted and superseded data, are given a time-stamp. All LADM classes (direct or indirect) inherit from VersionedObject (except for LA_Source). In this way, the contents of the database can be reconstructed, as they were at any historical moment. Objects begin, change and end due to events, which are represented in the source object LA_Source (which can be administrative, spatial or integrated).

Fig. 8 illustrates how the attributes *beginLifespanVersion* and *endLifespanVersion* are defined in the class VersionedObject. The LA_Source class has a *lifeSpanStamp* attribute and represents the event causing the changes in the registration. All the dates and times are system (or database) time, corresponding to the moment the event was processed and stored in the system. Constraints make sure the dates and times in LA_VersionedObject and LA_Source correspond. In addition VersionedObject and LA_Source have a second set of optional temporal attributes (*beginValidLifespanVersion*, *endValidLifespanVersion*, and *acceptance*), which represent to the corresponding valid times in the real world (according the source document/ event).

4.11. Status based history in LADM

For compatability with version 1 of the LADM, the proposed lifespan versioning is implemented as date/time stamps and the sID of LA_Source plays the role of event-id as described in section 4.5. This means that there is there is a reference to the event (LA_Source with CI_ResponsibleParty and DQ_Element), which was responsible for the database update that created this version, and to a quality statement.

4.12. Update mechanism in the bi-temporal model

The actual mechanism for updating bi-temporal objects is based on the versioned object model in LADM version 1. That is to say, rather than a database row being updated, it is flagged with a Tmax value (indicating "now") – effectively retiring that row. A new row is created with a Tmin value of "now" and copying the original Tmax value. This is the same as with the well-known pattern with a single timeline, and has the same constraints imposed (no overlapping of durations of the same object, co-incidence of adjacent durations of the same object, matching of LA-Source etc, but in this case, the constraints apply in both of the timelines. The example in section 4.8 illustrates the method of updating an object by retiring a row, and creating a new one.

The only additional complexity here is that it must be specified by the user whether it is a real-world change or a database correction, but this is an area for future research.

4.13. Event based history in LADM

The LA_Source class represents the event and contains a number of temporal attributes:

1 **lifeSpanStamp:** the TransactionEvent date/time of the beginning of the lifespan of the database record of all the (VersionedObject) object instances created by this event. This may also be the end of the lifespan of a set of object instances that were replaced by these instances. This is a Transaction Date event. ISO 19152: "The moment

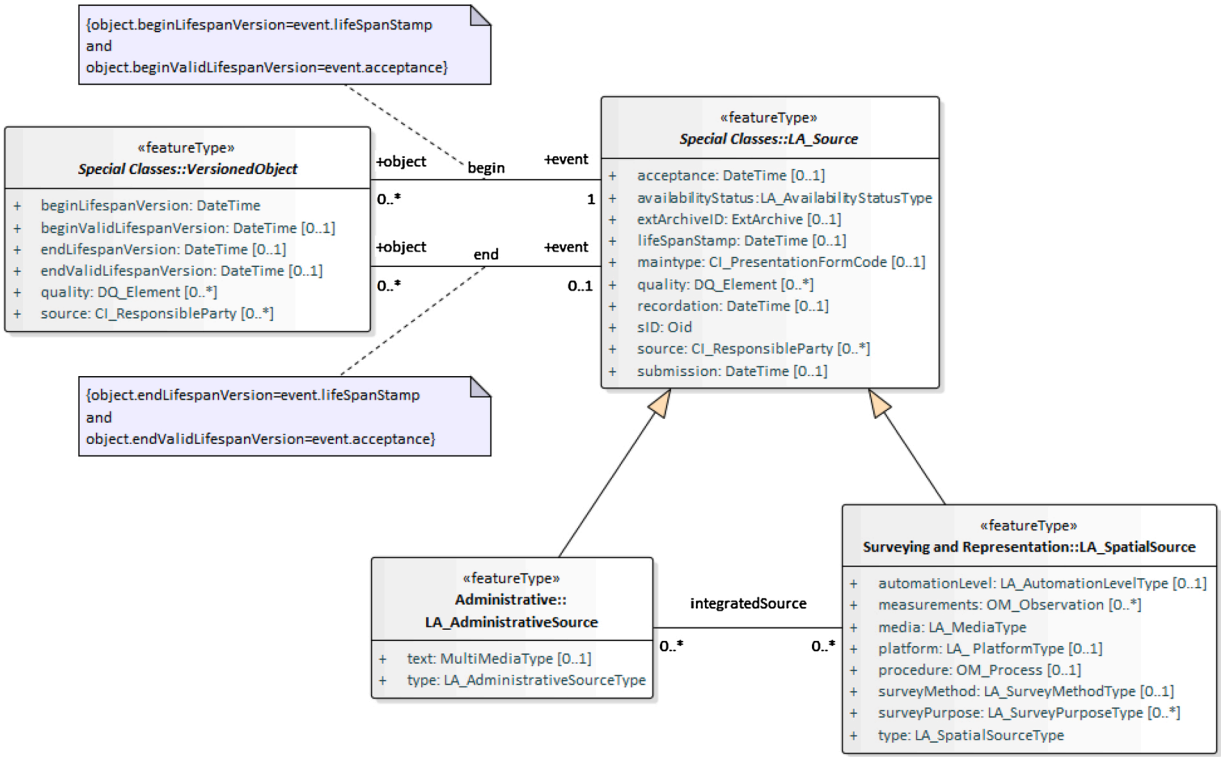


Fig. 10. The proposed integrated State and Event based LADM implementation of Bitemporal Objects.

that the event, represented by the instance of LA_Source, is further processed in the LA system (this is the moment of endLifespanVersion of old instances, and the moment of beginLifespanVersion of new instances)".

2 **acceptance:** the valid date/time of the event marking acceptance of the linked objects; e.g. this could be acceptance of the survey. ISO 19152: "The date of force of law of the source by an authority".

3 **recordation:** the date/time of the event indicating that the linked objects were recorded in the registry, not necessarily the database. ISO 19152: "The date of registration (recordation) of the source by the registering authority".

4 **submission:** the submission ValidEvent for the linked objects; e.g. this might be a survey plan submission. ISO 19152: "The date of submission of the source by a party".

A VersionedObject instance can never have more than one beginning source document, but may also have an ending source document. In the case of details of an object being added/corrected requiring a second source, then a new version of the object will be created and the old version marked as ended using the endLifespanVersion attribute. If available also the beginValidLifeSpanVersion and endValidLifeSpanVersion is maintained, by using the value of the acceptance attribute in the LA_Source event.

It is important to remember in all the cases described here (sections 4.5 to 4.12) that updates need only be recorded on VersionedObject object if that object actually changes as a result of that update. For example, if there is a change of ownership (recorded by a change in the LA_RRR - "Right Responsibility Restriction" information), but this change does not affect the shape of the spatial unit, there does not need to be a new instance of the LA_SpatialUnit record created.

5. Cadastral entities

The following point-like entities can exist in a cadastre, either as object classes or as attributes of other classes, depending on the design. The geometric attributes used to define these entities will depend on the SRS decision (See Section 3.1). Since most cadastral spatial units are defined as 2D, we begin with the point-like entities that are needed for a solely 2D Cadastre:

Vertex2D: A corner or any point-like entity along a cadastral boundary – the junction of 2 or -more lines in a 2D spatial unit boundary. It could also be the end of a hanging line in a not-fully-validated data set.

Node2D: A point-like entity at which three or more cadastral boundaries meet. A Node2D is a Vertex2D. Typically a Node2D will be used in the encoding of topological connectivity.

Knot2D, Centre2D, Focus2D etc.: Point-like entities that are used in the parametric definitions of curves, circles and ellipses etc.

TraversePoint2D: A Point-like entity used in a survey when the actual point positions cannot be reached (e.g. edges of marshland). (See points 9–13 and points 5 and 8 in Fig. 12)

SurveyMark2D: One of a series of physical objects that have been placed or identified to assist with the survey process. They may be permanent or temporary. (Examples: survey peg, screw in concrete, building corner, permanent survey marks).

Generally speaking these point-like entities in a cadastre represent fiat objects, and occupy a PlateFixedPoint. That is to say the entity moves with the Earth's crust. Some entities that define a natural ambulatory boundary are only defined at a point in time, and occupy a PointInstant - they are tangible. SurveyMarks are tangible objects, and occupy a PlateFixedPoint. If a mark is destroyed and replaced at the same location it may re-occupy the same PlateFixedPoint. It is quite possible for more than one point-like entity to occupy the same PlateFixedPoint, although this would be unusual (see Fig. 11).

Note that, even in a 2D cadastre, the Z value can be recorded and be significant. The distance between two points must allow for the fact that the vertical lines through two points diverge as Z increases. In a 2D Cadastre, the accuracy required of Z values is not high (tens of metres is OK), and a simple "approximate local ground level" will suit. Rather than record the Z value in the individual point records, an approximate DEM (Digital Elevation Model) or TIN can be used.

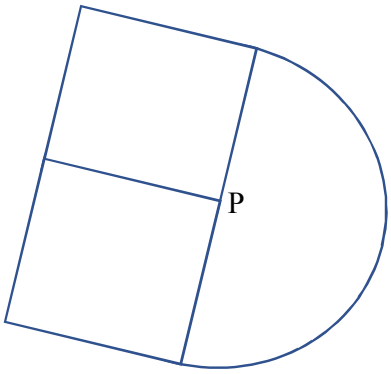


Fig. 11. Point P is a Node in the boundary of the two rectangular spatial units, but the same location is the centre of the circular boundary of the half-moon shaped spatial unit.

5.1. Database representation of 2D cadastral objects

For practical reasons, spatial databases today are based on a set of primitives such as "points", "lines" and "polygons", with differing topological intelligence. These primitives typically have the coordinates of the vertices of linear and area features embedded in the database representation of the features themselves.

It is clear from the above discussion that a point in a cadastre can carry a significant number of attributes beyond the simple (x, y, z) direct position. In particular, storing each cadastral corner as a Bitemporal Object may impose significant storage and response time overheads.

Some of the issues involved in abstracting the coordinate values of points from the geometric primitives are discussed in (Thompson, 2015), however that paper does not extend to bitemporal history.

5.2. Moving into 3D

The aim is to develop a Cadastral database containing both 2D and 3D spatial units (not a separate 3D cadastre). It should be clear that the Z direction is a special case in Cadastre. The most useful definition in Cadastre is that the (x, y) coordinate values along a vertical line do not vary. To a high accuracy, this is also parallel to the direction of gravity. For this reason, a useful approach to 3D Cadastre is to build on the existing 2D Cadastre entities.

It is well recognised (Stoter and van Oosterom, 2006) that a so-called 2D spatial unit actually defines a column of space above and below the ground level definition. The LADM defines the concept of LA_BoundaryFaceString (ISO-TC211, 2012), which is in effect a GM_MultiCurve extruded vertically (above and below the definition level). Thompson and van Oosterom (2012) define a number of terms to extend the LADM terminology of the "Boundary Face String" (Fig. 13). Thus, in Fig. 14, the 3D spatial unit is first defined in plan, and then the 3D details are provided in an isometric drawing. There is also a table of points – for example Vertex2D 19 has two 3D locations defined as 19a and 19b (circled in Fig. 14). The z values of these are presented in the table.

Point-like entities in 3D (see Fig. 13):

Pole: is equivalent to Vertex2D in a 3D context. It is a vertical line through the vertex (x_0, y_0), being the locus of all points (x, y, z) such that $x = x_0$, $y = y_0$, $-\infty < z < \infty$.

PlateFixedPole: The pole defined by a PlateFixedPoint (x_0, y_0, z_0) – the locus of all PlateFixedPoints (x, y, z) such that $x=x_0$, $y=y_0$, $-\infty < z < \infty$ (e.g. point 19 in Fig. 14).

Vertex3D: A corner or any point-like entity on a cadastral boundary

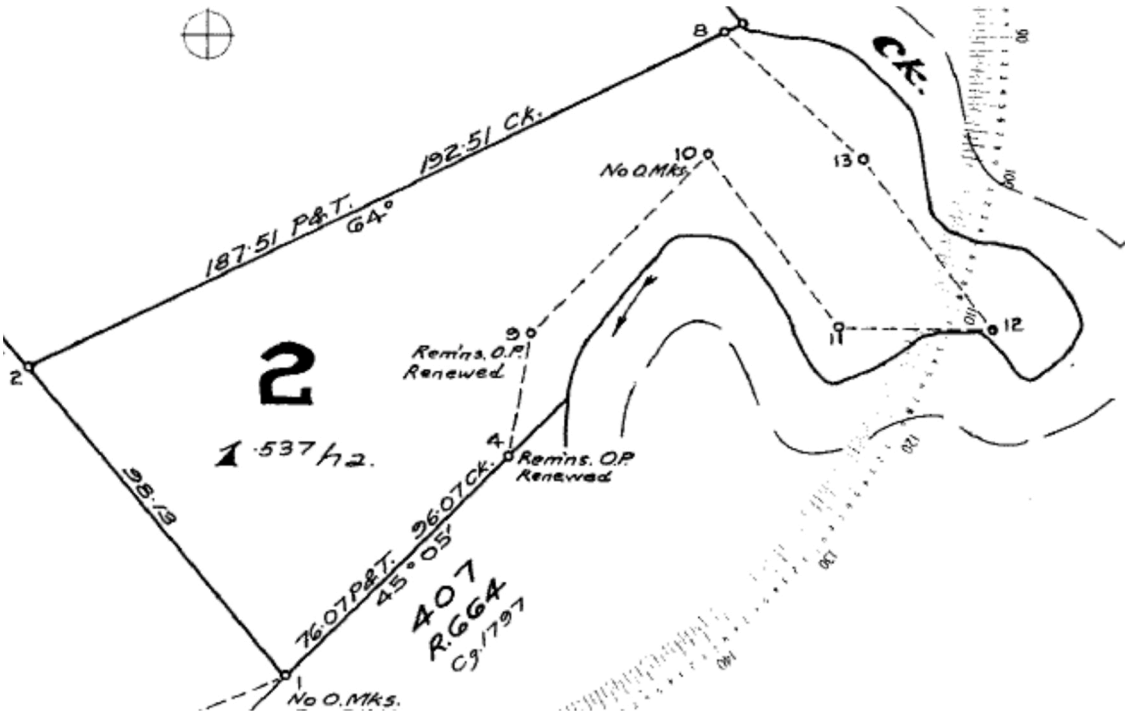


Fig. 12. Various point-like entities. Point 2 is a Node2D. Point 8 is a Vertex2D, but not a node in the cadastral fabric. Point 13 is a TraversePoint2D. The curved line is composed of a large number of Vertex2D objects which are on a natural boundary at an instant of time (in this case the plan date 9th Feb 1981).

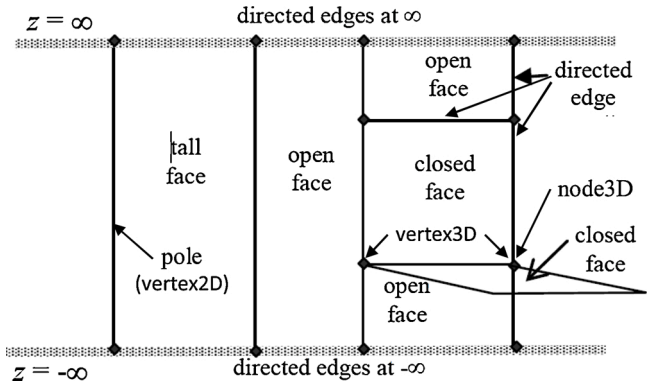


Fig. 13. Boundary face string concepts. This represents the side view of a boundary face string," flattened" into 2D for ease of presentation.

– the junction of 2 or more lines in a face, the point of meeting of 3 or more faces. Could also be the end of a hanging line or the vertex of a hanging face in a not-fully-validated data set.

Node3D: A point-like entity at which three or more cadastral boundary faces and lines meet. A Node3D is a Vertex3D. A Node3D may be useful in the encoding of topological connectivity (e.g. points 19a and 19b in Fig. 14).

Knot3D, Centre3D, Focus3D etc.: Point-like entities that are used in the parametric definitions of curved surfaces, circles and ellipses etc.

TraversePoint3D: A Point-like entity used in a survey when the actual point positions cannot be reached, or to connect two points that cannot be directly measured.

SurveyMark3D: A permanent or temporary survey mark can indicate a Z value.

In the discussion of PlateFixedPoint - Section 3.1, the actual movements considered were in the horizontal directions. Donnelly, Crook et al. restricted discussion to mainly 2D. "This paper does not further consider the vertical reference frame, except to note that analysis is required to determine how deformation modelling should be incorporated into a vertical reference frame, if at all, given that many engineering applications require that vertical deformation is visible in measurements." (Donnelly, Crook et al. 2015 Page 236). In this paper, we assume that the cadastre is defined in relation to the earth's surface – plate movements below the surface are assumed not to affect the legal boundary, so we can proceed as if all plate velocities are independent of Z value. (If this approximation is not accepted, modification of the below argument will be necessary). To a good approximation, the velocities can also be considered independent of time. That is - let $\boldsymbol{v}(x, y)$ be a vector velocity function for any point instant $p(x, y, z, t)$; then $\boldsymbol{v}(x, y, z, t) = (v_x(x, y), v_y(x, y), v_z(x, y))$ defines the movement of the plate at point instant (x, y, z, t). Thus if a PlateFixedPoint P exists at instant t at point (x, y, z) in an ITRF (i.e. at point instant (x, y, z, t)) then at a later instant t', its coordinates in the ITRF will be:

$$x' = x + (t' - t)\, v_x(x, y),$$

$$x' = y + (t' - t)\, v_y(x, y),$$

$$z' = z + (t' - t)\, v_z(x, y)$$

5.3. *Database representation of volumetric objects*

The majority of spatial units in a cadastre are simple 2D parcels. As a result, many users will be happy to use 2D GIS-type technology to access the data. It is important that they are given a suitable representation of the 3D spatial units – for example in terms of a "footprint" so that they are warned of the presence of volumetric parcels. This has been discussed in terms of a storage approach that combines a footprint, and bounding faces (ISO-TC211, 2012; Cemellini, van Oosterom et al. 2020; Kalogianni et al., 2020a).

In a similar way to the approach introduced in Section 5.1, the point coordinates may be abstracted and carry attributes including a history. This does impose an update overhead, because quite small adjustments of the positions of points that define 3D objects can cause consistency

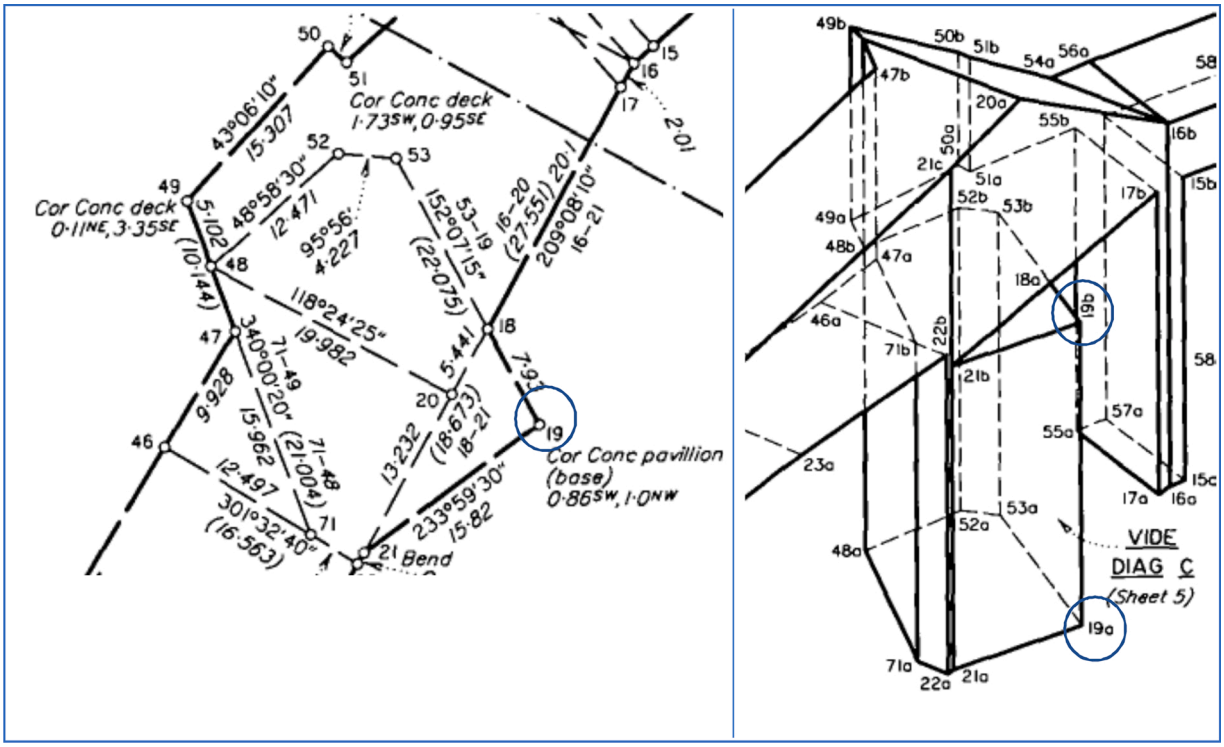


Fig. 14. A 3D spatial unit defined first in plan (L) then in an isometric view (R).

problems – e.g. moving the vertex of a 4 (or more) sided face may make it non-planar. This has been discussed, and concepts such as "robustness" and "flatness" suggested to ease this problem (Thompson and van Oosterom, 2006; Thompson, van Oosterom et al. 2019).

5.4. Representable point

It must be recognised that the coordinates cannot be directly represented as real numbers. They may be stored as floating or fixed-point approximations (discrete, finite precision). There are many calculations that introduce errors by virtue of the limited resolution. These must be considered, and it must be born in mind that two different PlateFixedPoints can become entangled by rounding errors.

5.5. Visualization of 3D bitemporal cadastre

There has been a significant amount of research into visualization of 3D cadastre - Shojaei (2014) identified 34 research projects on the issue. Recently the LADM approach to 2D and 3D integration has been researched from a visualization and management perspective (Cemellini, van Oosterom et al. 2020; Kalogianni et al., 2020a, Kalogianni et al., 2020b). By contrast, the query and visualization of the bi-temporal aspect of cadastre is a new field.

At present, there is no agreed set of requirements, but it is suggested that a successful "5D" visualization should have the following attributes:

- Ability to see the effect of a single event – perhaps using a "sweep line" or even two.
- Ability to see the evolution of a region – using one or two date/time sliders.
- Interaction of "pan and zoom" operations with temporal selection.
- Textural details of a specific event, including all objects affected.
- Textural history of a specific object (either valid events only or full bitemporal).

This is beyond the scope of this paper, and should be considered as a potentially rich field of future research. It must always be remembered here that most users will require access to the data using simple 2D software, and will only need to see the current data (see Section 5.3).

6. Conclusions

Existing definitions, and suggestions for terms for basic concepts relating to point-like objects (real-world, and representations) have been discussed in relation to the modification of information on those objects in a database. These have been further explored in terms of the concept of bi-temporal history, with particular reference to the LADM. The extension of these techniques to extended cadastral objects and into three spatial dimensions has been considered.

6.1. Future research

More research needs to be done to determine how to visualize and present 2D and 3D data with bitemporal support. But further, how to make it available to a wide range of users with a wide range of sophistication (and software) is an issue deserving significant research (see Section 5.5); as is the issue of updating the database while capturing the distinction between real-world and database changes (Section 4.11).

There is also additional research required to determine the complexity of data flows in a land administration system. For example, the registration of a plan of survey creates ValidEvents such as the making of the survey itself, the registration, and others; and TransactionEvents such as when it is entered to the database. In addition, if it creates the opportunity to improve the database accuracy, TransactionEvents can be applied to unrelated, nearby points. To add to the potential complexity, some jurisdiction could specify that a registration does not take effect until it is recorded in the database.

As has been indicated in section 4.7, maintaining database integrity in the presence of out-of-sequence ValidTime updates is a subtle problem – to avoid lost updates and topological failure in the historic record. The case where an attempt is being made to back-capture a full history of the cadastre is particularly complex. It is hoped that removal and re-

entry of all later (Valid) events can be avoided, and an attempt can be made to fit the historic event(s) into the current database record. In any case, the new historic versions of the involved objects will be marked with sysTmin/max as "now" and (Valid) Tmin/max as the historic date. This should be investigated further. It may be that capture of the history in reverse chronological order may help alleviate this problem.

One of the major advantages of the bitemporal approach to history is that it includes an audit trail of changes to the database that is accessible in terms of individual properties, parties and transactions. There is potential to use this in fraud detection and prevention, especially necessary where the data are made available on-line to the public (see example in Section 4.8).

It has been identified that there are benefits in abstracting individual points as versioned objects – rather than the common approach of holding their coordinates within spatial primitives, using the OGC Simple Feature Specification (OGC, 1999). This will probably incur a performance penalty, which should be quantified.

Declaration of Competing Interest

The authors have no conflicting interests in relation to this paper.

References

Burrough, P.A., 1986. Principles of geographical information systems for land resources assessment. Geocarto Int. 1 (3), 54.

Cemellini, B., van Oosterom, P., Thompson, R., de Vries, M., 2020. Design, development and usability testing of an LADM compliant 3D Cadastral prototype system. Land Use Policy.

Clark, D., 2015. Historical Records With PostgreSQL, Temporal Tables and SQL:2011. Retrieved Feb 2019, from. http://clarkdave.net/2015/02/historical-records-with-postgresql-and-temporal-tables-and-sql-2011/.

Couclelis, H., 1992. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In: Frank, A.U., Campari, I., Formentini, U. (Eds.), Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Springer-Verlag, Berlin.

Donnelly, N., Crook, C., Stanaway, R., Roberts, C., Rizos, C., Haasdyk, J., 2015. A Two-Frame National Geospatial Reference System Accounting for Geodynamics. IAG Commission 1 Symposium 2014: (REFAG2014) Reference Frames for Applications in Geoscience and Technology. Luxembourg, International Association of Geodesy.

ESRI, 2016. "Information Model Dictionary" Local Government Information Model. Retrieved Feb 2019, from. http://desktop.arcgis.com/en/arcmap/10.3/manage-data/editing-parcels/lgimdatadictionary.htm.

Frank, A.U., 1995. The prevalence of objects with sharp boundaries in GIS. In: Burrough, P.A., Frank, A.U. (Eds.), Geographic Objects With Indeterminate Boundaries. Taylor & Francis.

Grosvenor, 2019. Digital Survey Plans Review. Grosvenor Public Sector Advisory Final Report (v4.0), 17 December 2019.

ICSM, 2009. Geocentric Datum of Australia Technical Manual (Version 2.3, Amendment 1). Intergovernmental Committee on Surveying and Mapping.

ICSM, 2018. Geocentric Datum of Australia 2020 Technical Manual (version 1.2). Intergovernmental Committee on Surveying and Mapping.

ICSM, 2020. GDA Transformation Products and Tools. https://www.icsm.gov.au/datum/gda-transformation-products-and-tools.

ISO-TC211, 2002. ISO 19108: Geographic Information — Temporal Schema. International Organization for Standardization.

ISO-TC211, 2003. ISO 19103: Geographic Information - Spatial Schema. ISO19107, from 2001-2011-21. http://www.iso.org/iso/catalogue_detail.htm?csnumber=26012.

ISO-TC211, 2012. ISO19152, Geographic Information — Land Administration Domain Model (LADM).

Janée, G., Mathena, J., Frew, J., 2008. A data model and architecture for long-term preservation. 8th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL).

Jensen, C.S., Clifford, J., Elmasri, R., Gadia, S.K., Hayes, P., Jajodia, S., Dyreson, C., Grandi, F., Kafer, W., Kline, N., Lorentzos, N., Mitsopoulos, Y., Montanari, A., Nonen, D., Peressi, E., Pernici, B., Roddick, J.F., Sarda, N.L., Scalas, M.R., Segev, A., Snodgrass, R.T., Soo, M.D., Tansel, A., Tiberio, P., Wiederhold, G., 1994. A consensus glossary of temporal database concepts. ACM SIGMOD 23 (1), 52–64.

Jernigan, K., Guo, L., Krishnaswamy, V., Radhakrishnan, V., Raja, V., Shetler, T., 2009. Oracle Total Recall With Oracle Database 11g Release 2. from http://www.oracle.com/us/products/total-recall-whitepaper-171749.pdf..

Kalogianni, E., Dimopoulou, E., Thompson, R.J., Lemmen, C., Ying, S., Oosterom, P.V., 2020a. Development of 3D spatial profiles to support the full lifecycle of 3D objects. Land Use Policy 98.

Kalogianni, E., van Oosterom, P., Dimopoulou, E., Lemmen, C., 2020b. 3D land administration: a review and a future vision in the context of the spatial development lifecycle. Int. J. Geo Inf. 9 (107), 27.

Kulkarni, K., Michels, J.-E., 2012. Temporal features in SQL: 2011. Acm Sigmod Rec. 41 (3), 34–43.

McGarva, G., Morris, S., Janée, G., 2009. Technology watch report: preserving geospatial data. Digital Preservat. Coal. 33. May 2009.

OGC, 1999. Open GIS Simple Features Specification for SQL." Revision 1.1. Retrieved 15th Oct 2003. from. http://www.opengis.org/specs/.

Priebbenow, R., 1993. Survey infrastructure for Queensland. Survey Integration - the Future, Seminar and Workshop.

Saracco, C., Nicola, M., Gandhi, L., 2013. A Matter of Time: Temporal Data Management in DB2 10. from https://developer.ibm.com/articles/dm-1204db2temporaldata/..

Shojaei, D., 2014. 3D Cadastral Visualisation: Understanding Users' Requirements. PhD Thesis. University of Melbourne.

On drawing lines on a map. In: Smith, B. (Ed.), 1995. Spatial Information Theory. Lecture Notes on Computer Science. Springer, Berlin.

Snodgrass, R.T., Böhlen, M.H., Jensen, C.S., Steiner, A., 1998. Transitioning temporal support in TSQL2 to SQL3. Lecture Notes Comput. Sci. 1399, 150–194.

Stoter, J., van Oosterom, P., 2006. 3D Cadastre in an International Context. Boca Raton FL, Taylor & Francis.

Sweetkind-Singer, J., Larsgaard, M.L., Erwin, T., 2006. Digital preservation of geospatial data. Libr. Trends 55 (2), 304–314.

Tarbit, S., Thompson, R.J., 2006. Future trends for modern DCDB's, a new vision for an existing infrastructure. In: Combined 5th Trans Tasman Survey Conference and 2nd Queensland Spatial Industry Conference. Cairns, Queensland, Australia.

Thompson, R.J., 2015. A model for the creation and progressive improvement of a digital cadastral data base. Land Use Policy 49, 565–576.

Thompson, R.J., van Oosterom, P., 2006. Interchange of spatial data – inhibiting factors. In: 9th AGILE International Conference on Geographic Information Science. Visegrád, Hungary.

Thompson, R., van Oosterom, P., 2012. Validity of mixed 2D and 3D cadastral parcels in the Land administration domain model. In: 3rd International Workshop on 3D Cadastres: Developments and Practices. Shenzhen, China, pp. 325–344.

Thompson, R., Van Oosterom, P., 2019. A suggested terminology for Point-like entities in a Bi-temporal representation of 2D and 3D Land administration. In: 8th International FIG Workshop on the Land Administration Domain Model. P. Van Oosterom, C. Lemmen and A. A. Rahman. Kuala Lumpur, Malaysia, International Federation of Surveyors (FIG).

Thompson, R.J., van Oosterom, P., Karki, S., 2019a. Towards an Implementable Data Schema for 4D/5D Cadastre Including Bi-Temporal Support. FIG Working Week 2019, Hanoi, Vietnam.

Thompson, R.J., van Oosterom, P., Karki, S., 2019b. Towards an Implementable Data Schema for 4D/5D Cadastre Including Bi-Temporal Support. FIG Working Week 2019: Geospatial information for a smarter life and environmental resilience, Hanoi, Vietnam.

van Oosterom, P., 1997. Maintaining consistent topology including historical data in a large spatial database. Auto Carto 13. Seattle, WA.

Zadeh, L.A., 1974. Fuzzy logic and its application to approximate reasoning. Inf. Process. 74 (3), 591.

Rod Thompson has been working in the spatial information field since 1985. He designed and led the implementation of the Queensland Digital Cadastral Data Base. He obtained a PhD at the Delft University of Technology in December 2007.

Peter van Oosterom obtained an MSc in Technical Computer Science in 1985 from Delft University of Technology, the Netherlands. In 1990 he received a PhD from Leiden University. From 1985 until 1995 he worked at the TNO-FEL laboratory in The Hague. From 1995 until 2000 he was senior information manager at the Dutch Cadastre, where he was involved in the renewal of the Cadastral (Geographic) database. Since 2000, he is professor at the Delft University of Technology, and head of the 'GIS Technology' Section, Department OTB, Faculty of Architecture and the Built Environment, Delft University of Technology, the Netherlands. He is the current chair of the FIG Working Group on '3D Cadastres'. He is coeditor of the International Standard for the Land Administration Domain, ISO 19152.